AD-A219 194

(2)

AD

TECHNICAL REPORT ARCCB-TR-90005

ADAPTIVE MESH EXPERIMENTS FOR HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS

DAVID C. ARNEY

RUPAK BISWAS

JOSEPH E. FLAHERTY

FEBRUARY 1990

US ARMY ARMAMENT RESEARCH,
DEVELOPMENT AND ENGINEERING CENTER
CLOSE COMBAT ARMAMENTS CENTER
BENÉT LABORATORIES
WATERVLIET, N.Y. 12189-4050

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DISCLAIMER

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARCCB-TR-90005	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ADAPTIVE MESH EXPERIMENTS FOR HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David C. Arney, Rupak Biswas, and Joseph E. Flaherty (see reverse)		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army ARDEC Benet Laboratories Watervliet, NY 12189-4050		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AMCMS No. 6111.02.H610.011 PRON No. 1A84Z8CANMSC
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army ARDEC Close Combat Armaments Center Picatinny Arsenal, NJ 07806-5000		12. REPORT DATE February 1990 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Presented at the Sixth Army Conference on Applied Mathematics and Computing, University of Colorado, Boulder, CO, 31 May - 3 June 1988. Published in Proceedings of the Conference.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive Methods
Local Mesh Refinement
Hyperbolic Differential Equation
Error Estimation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We discuss experiments conducted on mesh moving and local mesh refinement algorithms that are used with a finite difference scheme to solve initial-boundary value problems for vector systems of hyperbolic partial differential equations in one dimension. The mesh moving algorithms move a coarse base mesh by a mesh movement function to follow and isolate spatially distinct phenomena. The local mesh refinement method recursively divides the time step and spatial cells in regions where error indicators are high until a prescribed

(CONT'D ON REVERSE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

7. AUTHORS (CONT'D)

David C. Arney
Department of Mathematics
United States Military Academy
West Point, NY 10996-1786

Rupak Biswas
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Joseph E. Flaherty
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

and

U.S. Army ARDEC
Close Combat Armaments Center
Benet Laboratories
Watervliet, NY 12189-4050

20. ABSTRACT (CONT'D)

error tolerance is satisfied.

The adaptive mesh algorithms are implemented in a code with an initial mesh generator, a MacCormack finite difference scheme, and an error estimator. Experiments are conducted for several different problems to determine the efficiency of the adaptive methods and their combinations and to gauge their effectiveness in solving one-dimensional problems.

NTIS	✓
By	
Dist	
A-1	

TABLE OF CONTENTS

TABLES

LIST OF ILLUSTRATIONS

INTRODUCTION

Our goal is to develop expert systems software for solving time-dependent partial differential equations. The software should allow users to describe problems in a natural language, have a convenient geometric description interface, and not require knowledge of sophisticated numerical analysis. The systems should be intelligent, efficient, reliable, robust, and able to solve a large class of problems to prescribed error tolerances.

The power of adaptive techniques is that they are capable of making decisions that change the computational environment. This significantly minimizes the number of a priori decisions demanded of the user and provides dramatic savings in the cost of the computation. This capability is performed by procedures that monitor intermediate results and feed this data back to a control mechanism that modifies the solution strategy. Three popular adaptive techniques for solving partial differential equations are mesh moving or rezoning (r-refinement), mesh refinement (h-refinement), and order enrichment (p-refinement). In r-refinement, the mesh is moved either continuously or statically at discrete times in order to resolve nonuniformities and reduce errors. H-refinement involves the addition or deletion of computational cells to the mesh and p-refinement involves increasing or decreasing the order of a method in different portions of the domain. All strategies attempt to organize the computation so that little effort is expended in regions where the solution is smooth and a much greater effort is devoted to regions where the solution is more difficult to compute.

The different refinement strategies are being combined to yield remarkable results. Babuska and Szabo (ref 1) showed that an hp-refinement scheme produced

an exponential rate of convergence on a singular elasticity problem. Arney and Flaherty (ref 2) developed an hr-refinement scheme that moved a 'base' coarse mesh to follow important dynamic structures of the solution and recursively refined the base mesh to improve resolution. They found that mesh motion was inexpensive relative to mesh refinement and reduced dispersive errors associated with wave motion but did not always accurately follow structures, especially when interactions occurred, and could not dependably satisfy prescribed tolerances. Recursive mesh refinement can satisfy prescribed tolerances but involves more complicated data structures and greater care at coarse-fine mesh interfaces than r-refinement.

There are numerous other variations of the three adaptive strategies for time-dependent problems. For example, temporal refinement can be done globally to produce an adaptive method of lines strategy (refs 3,4) or locally in combination with the spatial refinement strategy (refs 5,6).

Accurate a posteriori error estimation is essential for codes that strive to satisfy user-prescribed error tolerances. Error estimation is often the most expensive part of an adaptive algorithm. Arney and Flaherty (ref 2) calculated the local discretization error at nodes of the mesh using an algorithm based on Richardson's (ref 7) extrapolation. This pointwise estimate can then be used to construct several global measures of the discretization error. The advantage of this method is that it can be used to find error estimates for any numerical scheme without explicitly knowing the exact form of the error. Details of this error estimate and its implementation on a moving mesh are discussed in Arney (ref 8) and Arney et al. (ref 9).

In this report, we apply Arney and Flaherty's (ref 2) adaptive mesh moving and refinement technique to one-dimensional hyperbolic systems. As described herein, their approach consists of moving a base mesh of quadrilateral cells to

isolate important spatial structures of the solution. Refinement, when needed, is performed within cells of coarser meshes. Solutions are generated by a MacCormack (ref 10) finite difference scheme, and local error estimates, which are used to control mesh motion and refinement, are computed by Richardson's (ref 7) extrapolation. Our goal is to quantify the relative costs and benefits of mesh motion and local mesh refinement. In this report, we present the results of computational experiments performed on three one-dimensional problems using several conventional and adaptive numerical procedures. The results obtained demonstrate both the potential and limitations of the adaptive algorithm. We have mixed results showing that the effects of mesh moving can be problem-dependent. Generally, mesh motion is effective for following an isolated structure, but much less so when structures interact. In this report, we also discuss the utility of our methods, the computational results, and future work.

ALGORITHM

We consider an application of Arney and Flaherty's (ref 2) adaptive procedure to one-dimensional vector systems of hyperbolic conservation laws having the form

$$\vec{u}_t + \vec{f}_x(x,\vec{u},t) = 0,\ x \in D,\ t > 0 \tag{1}$$

$$\vec{u}(x,0) = \vec{u}_0(x),\ x \in D \cup \partial D \tag{2}$$

with appropriate well-posed conditions on the boundary ∂D of a domain D. Like them, we discretize Eqs. (1) and (2) using a MacCormack (ref 10) finite difference scheme because of its general applicability (ref 11). Although this scheme suffers a reduction in order on a moving nonuniform grid, our computations show that proper mesh moving can provide enough efficiency and accuracy to compensate for this order reduction.

The MacCormack scheme produces spurious oscillations near discontinuities because it is a centered scheme with second order accuracy on a uniform mesh. The use of artificial viscosity to make this scheme total variation diminishing (TVD) makes it attractive as a general solver for problems with discontinuities and we use a model by Davis (ref 12). The artificial viscosity terms are calculated from the solution data at the beginning of each time step and are added to the solution after the MacCormack solution has been calculated.

Arney and Flaherty's (ref 13) mesh moving procedure is based on an intuitive approach that allows nodes to follow local nonuniformities rather than the more analytical approaches of equidistribution of error (ref 4) or the solving of variational problems to minimize some given functional (ref 14) which can be expensive and problem-dependent. They derive equations for the nodal velocities so that the mesh moves to follow the geometric propagation of some local nonuniformity. This generally reduces dispersive errors and allows the use of larger time steps while maintaining accuracy and stability. Important factors for mesh moving are to maintain mesh smoothness by controlling adjacent cell ratios, to keep nodes within the domain boundaries, and to move nodes with a velocity that reduces discretization error. In order to prevent mesh distortion that can lead to increased discretization error of the solver, mesh points cannot move independently but must be coupled to at least some of their neighbors.

Some schemes do this coupling by attraction and repulsion of nodes (cf. Rai and Anderson (ref 15)). In these algorithms, the coupling is done globally, where each node influences the velocity of all other nodes in the mesh. Attempting to equidistribute errors can lead to problems where nodes move incorrectly in some regions. This occurs, for example, when a mesh that is following one structure must react to another nonuniformity that arises in

another part of the domain. An abrupt grid adjustment can be eliminated if the influence is more local and the movement algorithm is combined with a mesh refinement scheme to add the necessary nodes in the region of the new structure.

At each time step, the selection mechanism of Arney and Flaherty's (ref 2) mesh moving algorithm uses the current node locations and the nodal values of a mesh movement indicator at the independent moving nodes of a coarse mesh as feedback. The local error estimates are used as the mesh movement indicators. Nodes with 'significant error' are grouped into error clusters. This clustering separates the important spatially distinct phenomena of the solution. As time evolves, the clusters can move, change size, collide, or separate. At each time step, new clusters can be created and old ones can vanish.

Mesh movement is then determined by each node's relationship to its nearest error cluster and the propagation velocity of the center of error mass of the cluster. Therefore, the nodal influence is regional. The amount of movement is determined by a movement function which insures that the center of error of the cluster moves according to a differential equation suggested by Coyle et al. (ref 16)

$$\ddot{r} + \lambda\dot{r} = 0 \tag{3}$$

where $r(t)$ is the position of the center of error mass of a cluster and $(\dot{\ }) := d()/dt$. Additionally, this movement function smoothes the mesh motion and prevents nodes from moving outside the domain boundary. The distance a node moves is reduced near boundaries to prevent it from leaving the domain. Nodes on the domain boundaries are not allowed to move.

Arney and Flaherty (ref 2) perform static rezoning whenever computation with the current mesh would be counterproductive or when the current mesh suffers from poor mesh ratios. There are sophisticated algorithms to check the

mesh condition and to verify the validity of the mesh (cf. Babuska and Szabo (ref 1) and Simpson (ref 17)). These algorithms can check for gaps between cells and overlapping cells. Since moving meshes can only develop such severe problems over time, mesh degradation can be discovered before it develops complete invalidity. This mesh degradation or ill-conditioning occurs when the mesh angles are severe, the mesh contains poor mesh ratios or poor aspect ratios, or the time step is too restrictive because of the crowding of nodes. Nodes of the coarse mesh can become too crowded when error clusters pass through boundaries or when two or more error clusters converge and trap nodes between them. Static rezoning is performed only when absolutely necessary due to the high cost in accurately interpolating the solution from the existing mesh onto a new one. It was not done in any of the examples presented in this report.

Arney and Flaherty (ref 2) used local mesh refinement to insure that the user-prescribed error tolerance was satisfied. This was done by recursively introducing finer meshes by binary refinement of space-time cells in regions where nodes with unacceptable error have formed clusters (cf. Berger (ref 18), Flaherty and Moore (ref 6), and Gropp (ref 19)). The clustering algorithm used for refinement is the same as the one used for mesh movement. The clusters are buffered so that high error nodes are in the interior of the refined region. The problem is recursively solved on these fine meshes until the error is within the specified tolerance. The refined subgrids that are adaptively created by the local refinement algorithm overlay the coarser grids. Each of these subgrids is independently defined. Figure 1 shows a coarse grid with portions overlayed by two fine grids and three finer grids.

Arney and Flaherty's (ref 5) mesh refinement strategy suggests the use of a tree data structure for its description and implementation. In this tree structure, the coarsest grid is the root node and is defined as level 0 in the tree.

The subgrids of the coarse grid are its offspring in the tree and are defined as level 1. A grid at level ℓ is properly nested in the tree between its parent at level $\ell - 1$ and its offspring (if any) at level $\ell + 1$. Grids at the same level are given an arbitrary ordering. Due to the clustering and buffering of error regions, grids at the same level of a two-dimensional problem can intersect and overlap. Figure 1 depicts an example of a sequence of meshes that might be produced by our refinement procedure for a coarse grid refinement step. The numbers next to the grids indicate the order in which the solution is computed on each grid.

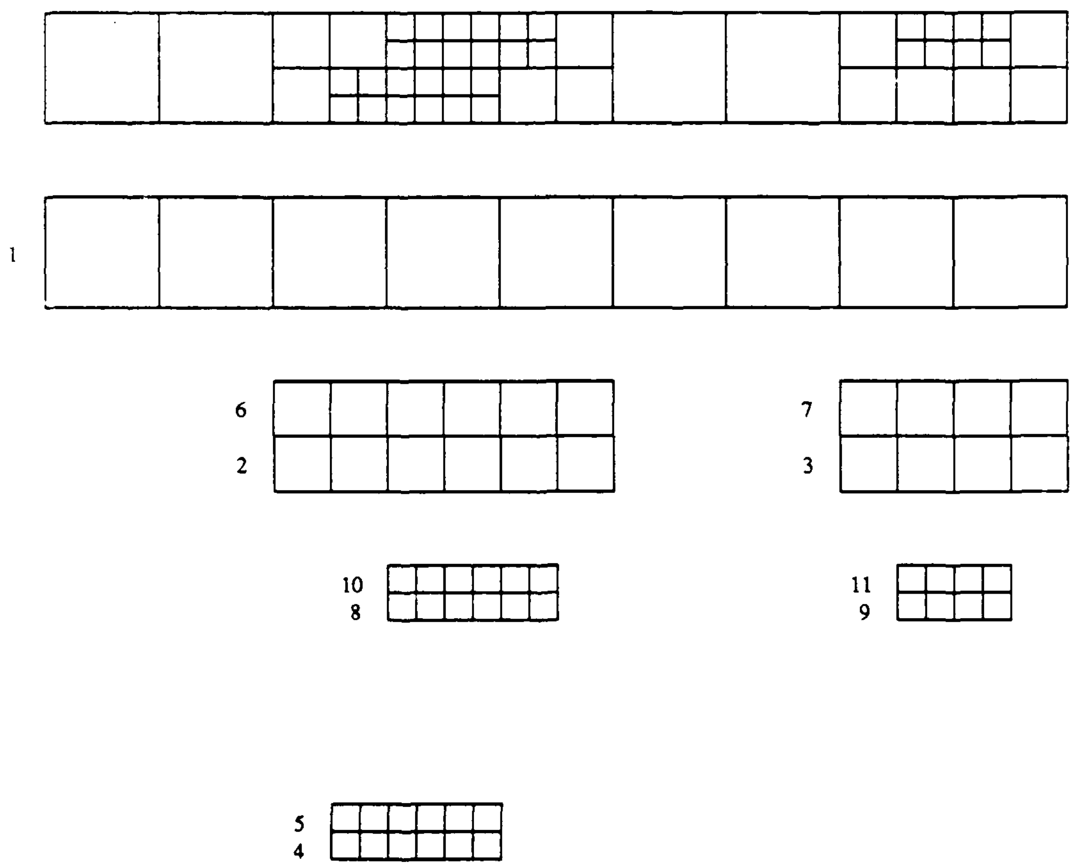

Figure 1. Typical set of local refinement grids for one coarse time step. The numbers indicate the order in which the solution is computed on each grid.

Such tree data structures are commonly used in adaptive mesh refinement procedures (cf. Berger and Oliger (ref 20) and Flaherty and Moore (ref 6)). Additionally, we use a stack to implement the recursive algorithm (cf. Aho et al. (ref 21) and Horowitz and Sahni (ref 22)).

The solution vectors, error estimates, and nodal information are all stored in a dynamic storage area with pointers from the tree to this storage area for each mesh in the tree. For each grid, we store its level in the tree and the number of nodes it contains. The dynamic storage area contains the solution vector and the error estimate at each node, and nodal information for use in the solver and grid interface procedures. Since the old mesh data is saved to obtain initial data for newly refined grids and nodes of the parent grids are updated from the fine grid solutions, nodal relationships between meshes are stored directly in the nodal vector.

COMPUTATIONAL RESULTS

We conducted experiments of Arney and Flaherty's (ref 2) adaptive mesh strategy using three problems and our results follow. In each case, errors are measured in the L_1 norm and the CPU times are normalized to unity. All calculations were performed on an IBM 3081D computer.

Example 1

Consider the scalar hyperbolic differental equation

$$u_t + (\cos \pi t)\, u_x = 0, \quad t > 0, \quad -0.4 \leqslant x \leqslant 1.4 \tag{4}$$

with initial conditions

$$u(x,0) = \begin{cases} 1, & \text{if } 0.4 \leqslant x \leqslant 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

and boundary conditions

$$u(-0.4,t) = u(1.4,t) = 0. \tag{6}$$

The exact solution to this problem is

$$u(x,t) = \begin{cases} 1, & \text{if } 0.4 \leqslant x - (\sin \pi t)/\pi \leqslant 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

which is a square pulse of unit amplitude that oscillates sinusoidally about the center of the domain. Artificial viscosity was added to eliminate oscillations in the solution; however, this resulted in an attentuation and spreading of the square pulse.

Four different adaptive strategies were used to solve this problem for $0 \leqslant t \leqslant 2.5$. The solutions at several times, the mesh trajectories, and the time step profile for the various strategies are shown in Figures 2, 3, 4, and 5. Table I summarizes the comoutational cost and accuracy of the four strategies.

TABLE I. COMPARISON OF THE DIFFERENT ADAPTIVE STRATEGIES FOR EXAMPLE 1

Adaptive Strategy	$\|e\|_1$	Number of Space-Time Cells	Normalized CPU Time	Attentuation
1. Stationary uniform mesh	0.1090	774	1.000	0.545
2. Moving mesh	0.0903	1134	1.452	0.730
3. Stationary uniform mesh with refinement	0.0614	15718	8.761	0.969
4. Moving mesh with refinement	0.0395	16554	10.069	0.994

With a stationary uniform mesh, we find that the square pulse is rapidly attenuated and diffused. The time step profile shows how the Courant number is utilized to maintain maximum step sizes without loss of stability. From Figure 3, it is apparent that the results improve when the mesh is allowed to move.

The pulse is attenuated less and the error is reduced, but more time steps are needed to complete the computation. The mesh trajectories in Figure 3 demonstrate how well the nodes track the square pulse as it oscillates. Figures 4 and 5, depicting the results of strategies 3 and 4, respectively, show remarkable improvement when adaptive mesh refinement is used. In both cases, the local error tolerance was specified as 0.001. Errors are reduced and the attenuation of the pulse is almost negligible, but shape distortion is still significant. Notice how well the refinement procedure tracks the pulse; however, the cost of computation increases by almost an order of magnitude. When moving and refinement are combined, the results are even more remarkable. The pulse is attenuated by a factor of only 0.6 percent.

Example 2

Consider the linear uncoupled system

$$\begin{aligned} u_t + (\cos \pi t)u_x &= 0, \\ v_t - (\cos \pi t)v_x &= 0, \end{aligned} \qquad t > 0, \quad -0.4 \leqslant x \leqslant 1.4 \tag{8}$$

with initial conditions

$$u(x,0) = v(x,0) = \begin{cases} 1, & \text{if } 0.4 \leqslant x \leqslant 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

and boundary conditions

$$u(-0.4,t) = u(1.4,t) = v(-0.4,t) = v(1.4,t) = 0. \tag{10}$$

The exact solution to this problem is

$$u(x,t) = \begin{cases} 1, & \text{if } 0.4 \leqslant x - (\sin \pi t)/\pi \leqslant 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$v(x,t) = \begin{cases} 1, & \text{if } 0.4 \leqslant x + (\sin \pi t)/\pi \leqslant 0.6 \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The first component u is the same as in Example 1, and the second component

v moves symmetrically with u. Four different adaptive strategies were used to solve this problem for $0 \leq t \leq 1.5$. Table II summarizes the computational cost and accuracy of the four strategies. The solutions at several times, the mesh trajectories, and the time step profile for mesh strategies 3 and 4 are shown in Figures 6 and 7, respectively.

It is clear that mesh moving does not provide the expected improvement in the results for this problem. In fact, we can see from Table II that each time the mesh is moved, the error in the computed solution increases. This is because two identical error regions moving symmetrically about the center of the domain do not contribute equally to the mesh motion due to asymmetries in their error estimates. As a result, the mesh moves incorrectly and the solution deteriorates. This, in turn, leads to further imbalance of the error clusters and subsequently causes catastrophic effects. Comparing Figures 6 and 7, we see how bad the solution is attenuated due to incorrect mesh motion. Improper mesh motion has also led to refinement in some regions of the mesh where it should not have been necessary. In both cases, the local error tolerance was specified to be 0.005.

TABLE II. COMPARISON OF THE DIFFERENT ADAPTIVE STRATEGIES FOR EXAMPLE 2

Adaptive Strategy	$\|e\|_1$	Number of Space-Time Cells	Normalized CPU Time
1. Stationary uniform mesh	0.1145	1650	1.000
2. Moving mesh	0.1221	5640	3.386
3. Stationary uniform mesh with refinement	0.0541	20828	6.926
4. Moving mesh with refinement	0.0583	48954	18.667

Example 3

Consider the coupled hyperbolic system from the wave equation

$$\begin{aligned} u_t - v_x &= 0, \\ v_t - u_x &= 0, \end{aligned} \qquad t > 0, \qquad -0.3 \leqslant x \leqslant 1.4 \tag{13}$$

with initial conditions

$$u(x,0) = \begin{cases} 1, & \text{if } 0.4 \leqslant x \leqslant 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$v(x,0) = \begin{matrix} 1, & \text{if } 0.5 \leqslant x \leqslant 0.7 \\ 0, & \text{otherwise} \end{matrix} \tag{15}$$

and boundary conditions satisfying the exact solution

$$u(x,t) = (p(x+t) + q(x-t))/2.0 \tag{16}$$

$$v(x,t) = (p(x+t) - q(x-t))/2.0 \tag{17}$$

where

$$p(\xi) = \begin{cases} 2, & \text{if } 0.5 \leqslant \xi \leqslant 0.6 \\ 1, & \text{if } 0.4 \leqslant \xi < 0.5 \ \text{ or } \ 0.6 < \xi \leqslant 0.7 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

and

$$q(\xi) = \begin{cases} 1, & \text{if } 0.4 \leqslant \xi \leqslant 0.5 \\ -1, & \text{if } 0.6 \leqslant \xi \leqslant 0.7 \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Four different adaptive strategies were used to solve this problem for $0 \leqslant t \leqslant 0.6$. Table III summarizes the computational cost and accuracy of the four strategies. The solutions at several times, the mesh trajectories, and the time step profile for mesh strategies 3 and 4 are shown in Figures 8 and 9, respectively.

Once again, mesh motion does not appear to result in the desired improvement in the solution. In this case, there are two error regions moving away

with unit speed in opposite directions from the center of the domain. However, the error regions are not identical as was the case in Example 2. With a moving mesh, the solution is attenuated and consequently, the error measure in the L_1 norm increases.

TABLE III. COMPARISON OF THE DIFFERENT ADAPTIVE STRATEGIES FOR EXAMPLE 3

Adaptive Strategy	$\|e\|_1$	Number of Space-Time Cells	Normalized CPU Time
1. Stationary uniform mesh	0.1141	930	1.000
2. Moving mesh	0.1057	3180	3.454
3. Stationary uniform mesh with refinement	0.0527	23236	11.493
4. Moving mesh with refinement	0.0552	39234	20.232

CONCLUSION

We have experimented with the adaptive mesh method of Arney and Flaherty (ref 2). Our results indicate that proper mesh moving can efficiently reduce errors. However, their mesh moving is not effective for problems that have more than one moving structure. We find that whenever there are two error regions, the mesh moving strategy is unable to make an accurate decision. This occurs particularly during the time when the two structures have not completely separated but still form one large error cluster. Results of local refinement tests show that it can efficiently reduce errors. The most powerful method was the combination of both mesh moving and mesh refinement. Results obtained for Example 1 show that a totally adaptive mesh strategy can be extremely effective. The overhead associated with the clustering and dynamic data structures is only

about 5 percent of the time needed to calculate a comparable solution on a uniform mesh.

Additional computation is needed to verify the generality of these conclusions. It is also not clear how many of the difficulties were due to MacCormack's (ref 10) finite difference scheme or Richardson's (ref 7) extrapolation-based error estimate. A TVD scheme would greatly improve performance near discontinuities.

We are re-examining the entire process in order to determine an effective mesh procedure. Future computations will be performed using more advanced shock capturing difference schemes (e.g., Engquist and Osher (ref 23)).

REFERENCES

1. I. Babuska and B. Szabo, "On the Rates of Convergence of the Finite Element Method," Intl. J. Numer. Methods in Engrg., Vol. 18, 1982, pp. 323-341.

2. D.C. Arney and J.E. Flaherty, "An Adaptive Method With Mesh Moving and Local Mesh Refinement for Time-Dependent Partial Differential Equations," Transactions of the Fourth Army Conference on Applied Mathematics and Computing, ARO Report 87-1, U.S. Army Research Office, Research Triangle Park, NC, 1987, pp. 1115-1141; also ARDEC Technical Report ARCCB-TR-88017, Benet Laboratories, Watervliet, NY, April 1988.

3. S. Adjerid and J.E. Flaherty, "A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time-Dependent Partial Differential Equations," SIAM J. Numer. Anal., Vol. 23, No. 4, 1986, pp. 778-796.

4. S.F. Davis and J.E. Flaherty, "An Adaptive Finite Element Method for Initial-Boundary Value Problems for Partial Differential Equations," SIAM J. Sci. Stat. Comput., Vol. 3, 1982, pp. 6-27.

5. D.C. Arney and J.E. Flaherty, "An Adaptive Local Mesh Refinement Method for Time-Dependent Partial Differential Equations," Appl. Num. Math., Vol. 5, 1989, pp. 257-274.

6. J.E. Flaherty and P.K. Moore, "A Local Refinement Finite Element Method for Time-Dependent Partial Differential Equations," Transactions of the Second Army Conference on Applied Mathematics and Computing, ARO Report 85-1, U.S. Army Research Office, Research Triangle Park, NC, 1985, pp. 585-595; also ARRADCOM Technical Report ARLCB-TR-85028, Benet Weapons Laboratory, Watervliet, NY, August 1985.

7. L.F. Richardson, "The Deferred Approach to the Limit. Part I. - Single Lattice," Phil. Trans. Roy. Soc. London, Vol. 226, 1927, pp. 299-349.

8. D.C. Arney, "An Adaptive Mesh Algorithm for Solving Systems of Time-Dependent Partial Differential Equations," Ph.D. Thesis, Department of Mathematics, Rensselaer Polytechnic Institute, Troy, NY, 1985.

9. D.C. Arney, R. Biswas, and J.E. Flaherty, "A Posteriori Error Estimation of Adaptive Finite Difference Schemes for Hyperbolic Systems," Transactions of the Fifth Army Conference on Applied Mathematics and Computing, ARO Report 88-1, U.S. Army Research Office, Research Triangle Park, NC, 1988, pp. 437-457; also ARDEC Technical Report ARCCB-TR-88024, Benet Laboratories, Watervliet, NY, June 1988.

10. R.W. MacCormack, "Numerical Solution of the Interaction of a Shock Wave with a Laminar Boundary Layer," Proc. Second Intl. Conf. on Numer. Methods in Fluid Dynamics., (M. Holt, ed.), Lecture Notes in Physics, Vol. 8, Springer-Verlag, New York, 1971, pp. 151-163.

11. A.R. Mitchell and D.F. Griffiths, The Finite Difference Method in Partial Differential Equations, Wiley, Chichester, UK, 1980.

12. S.F. Davis, "A Simplified TVD Finite Difference Scheme via Artificial Viscosity," SIAM J. Sci. Stat. Comput., Vol. 8, 1987, pp. 1-18.

13. D.C. Arney and J.E. Flaherty, "A Two-Dimensional Mesh Moving Technique for Time-Dependent Partial Differential Equations," J. Comput. Phys., Vol. 67, 1986, pp. 124-144.

14. J. Djomehri and K. Miller, "A Moving Finite Element Code for General Systems of PDE's in 2-D," PAM-57, Center for Pure and Appl. Math., University of California, Berkeley, 1981, pp. 1-49.

15. M.M. Rai and D.A. Anderson, "Grid Evolution in Time Asymptotic Problems," J. Comput. Phys., Vol. 43, 1981, pp. 327-344.

16. J.M. Coyle, J.E. Flaherty, and R. Ludwig, "On the Stability of Mesh Equidistribution Strategies for Time-Dependent Partial Differential Equations," J. Comput. Phys., Vol. 62, 1986, pp. 26-39.

17. R.B. Simpson, "A Two-Dimensional Mesh Verification Algorithm," SIAM J. Sci. Stat. Comput., Vol. 2, 1981, pp. 455-473.

18. M.J. Berger, "Data Structures for Adaptive Mesh Refinement," Adap. Comp. Methods for Part. Diff. Eqns., (I. Babuska, J. Chandra, and J.E. Flaherty, eds.), SIAM, Philadelphia, 1983, pp. 237-251.

19. W.D. Gropp, "A Test of Moving Mesh Refinement for 2-D Scalar Hyperbolic Problems," SIAM J. Sci. Stat. Comput., Vol. 1, 1980, pp. 191-197.

20. M.J. Berger and J. Oliger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," J. Comput. Phys., Vol. 53, 1984, pp. 484-512.

21. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Massachusetts, 1974.

22. E. Horowitz and S. Sahni, Fundamentals of Data Structures, Computer Science Press, Woodland-Hills, California, 1976.

23. B. Engquist and S. Osher, "One-Sided Difference Approximations for Nonlinear Conservation Laws," Math. Comp., Vol. 36, 1981, pp. 321-351.

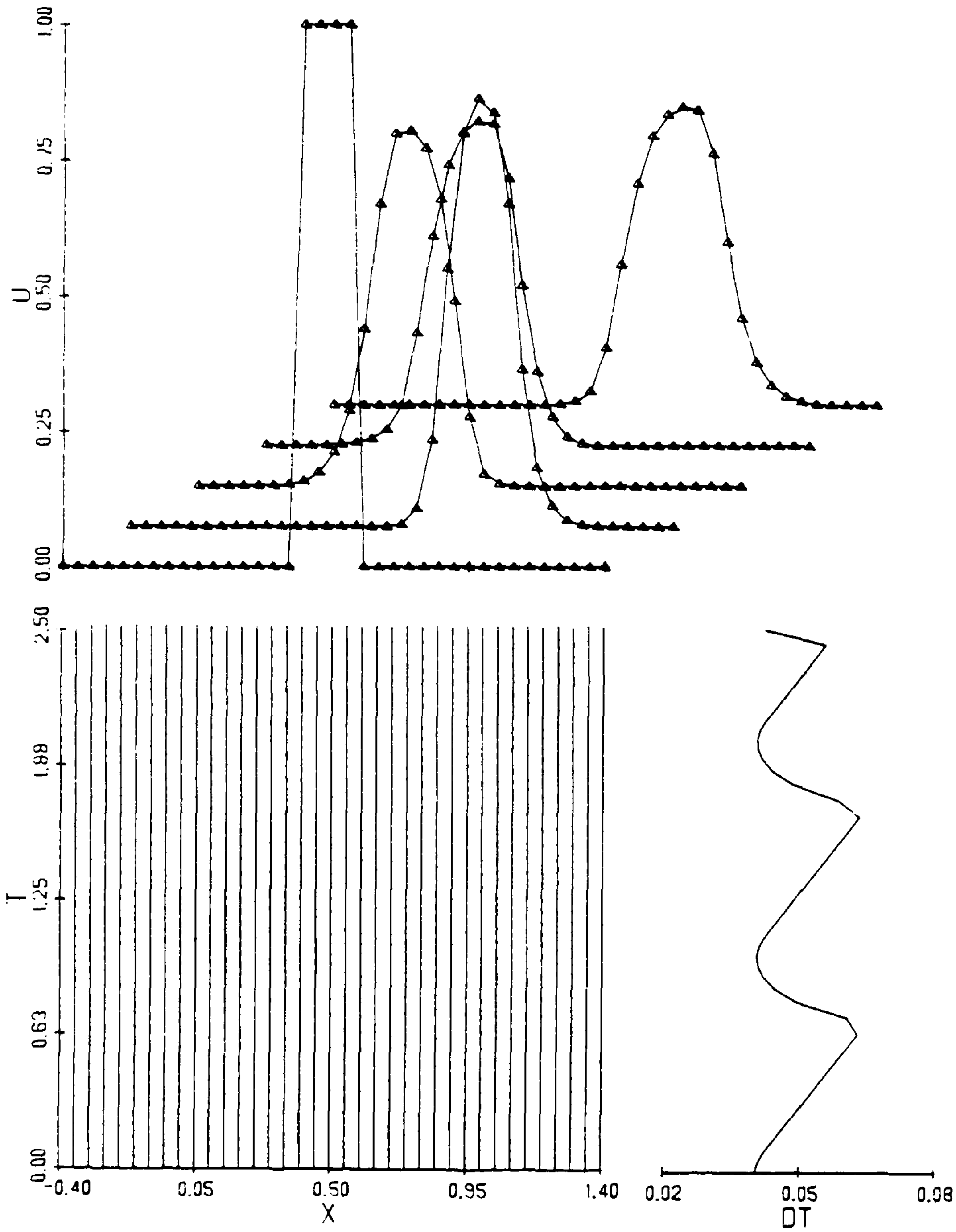

Figure 2. Solutions at t = 0, 0.57, 1.15, 1.80, and 2.33, mesh trajectories, and time step profile for strategy 1 of Example 1.

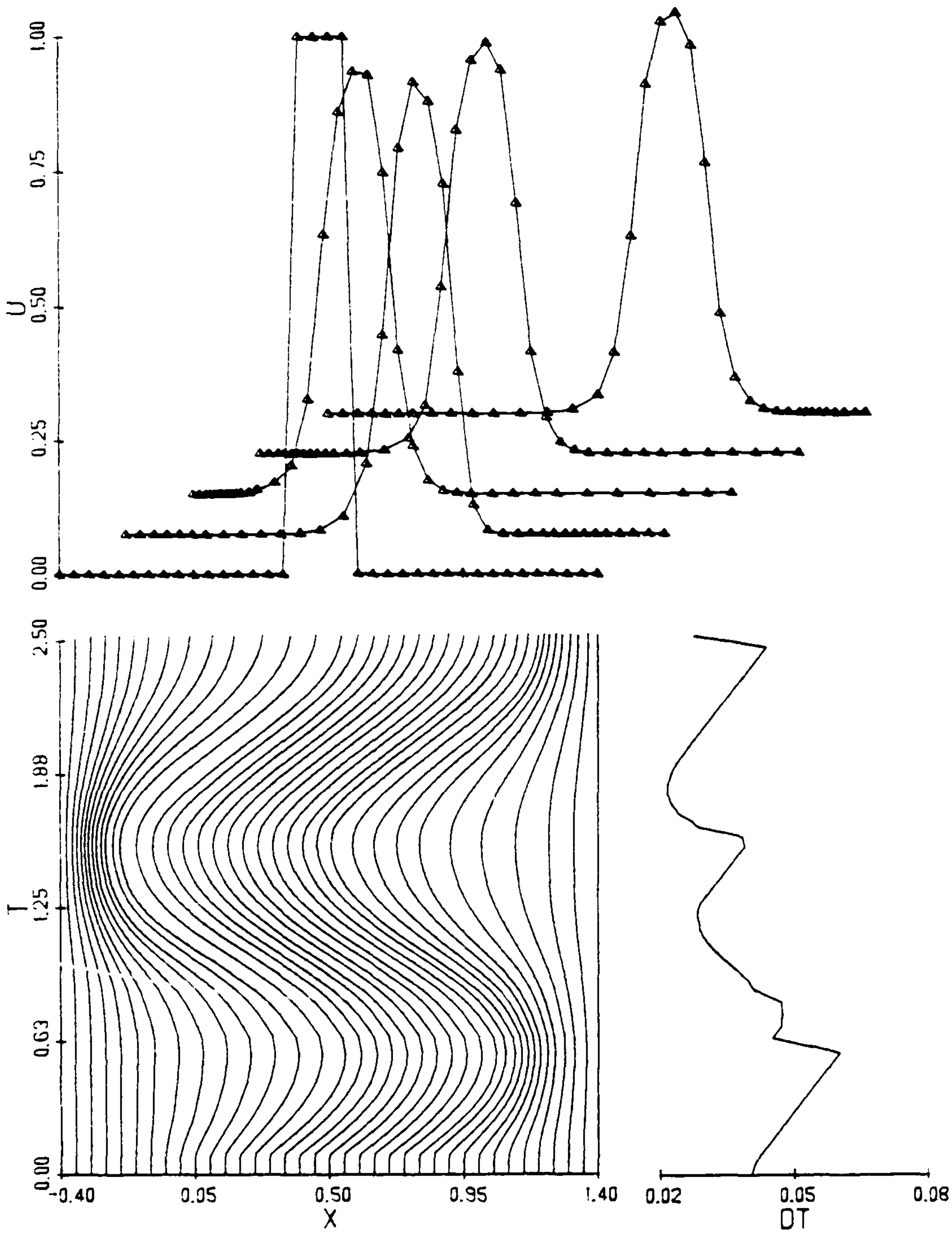

Figure 3. Solutions at t = 0, 0.87, 1.42, 1.88, and 2.37, mesh trajectories, and time step profile for strategy 2 of Example 1.

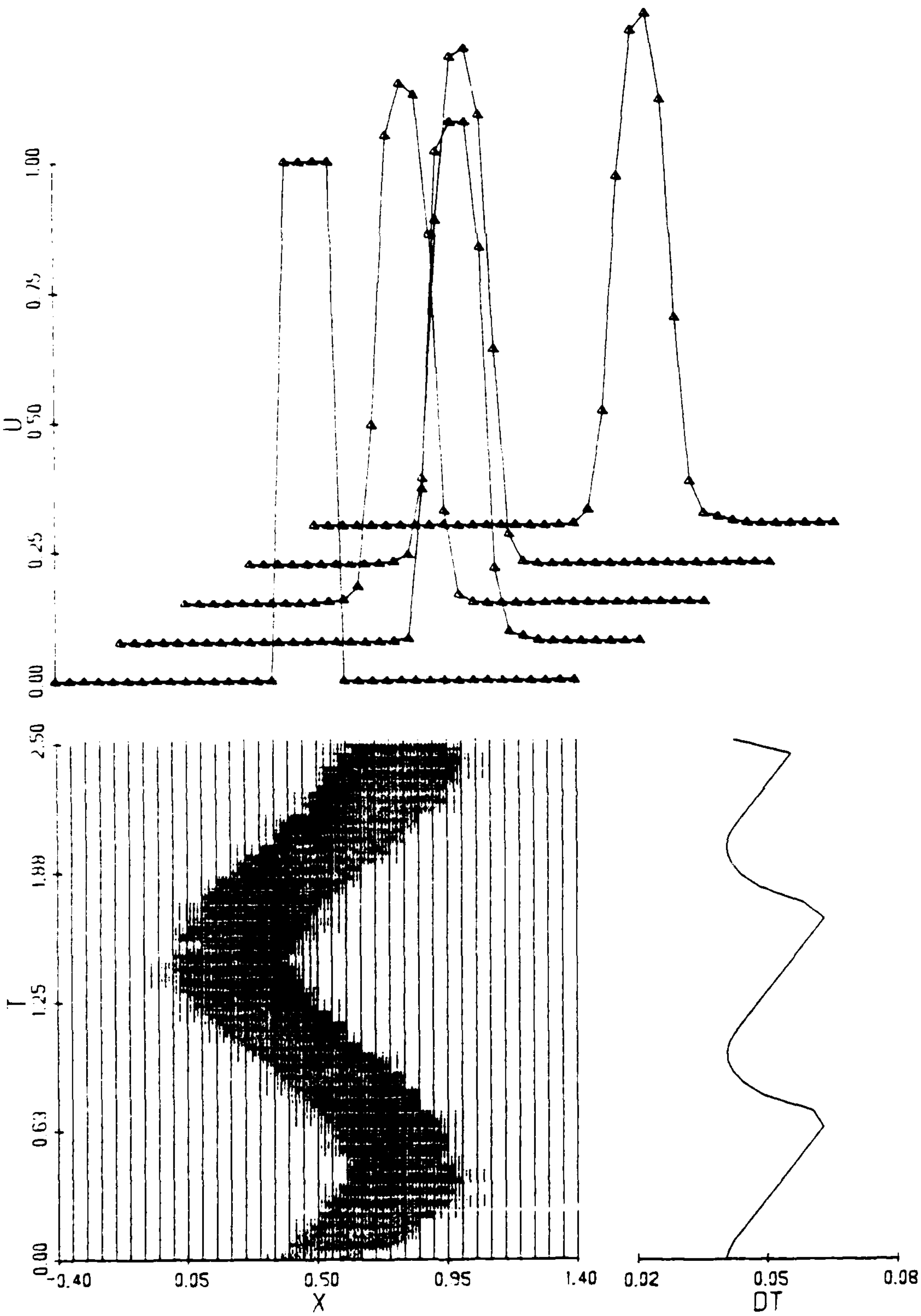

Figure 4. Solutions at t = 0, 0.57, 1.15, 1.80, and 2.33, mesh trajectories, and time steo profile for strategy 3 of Example 1.

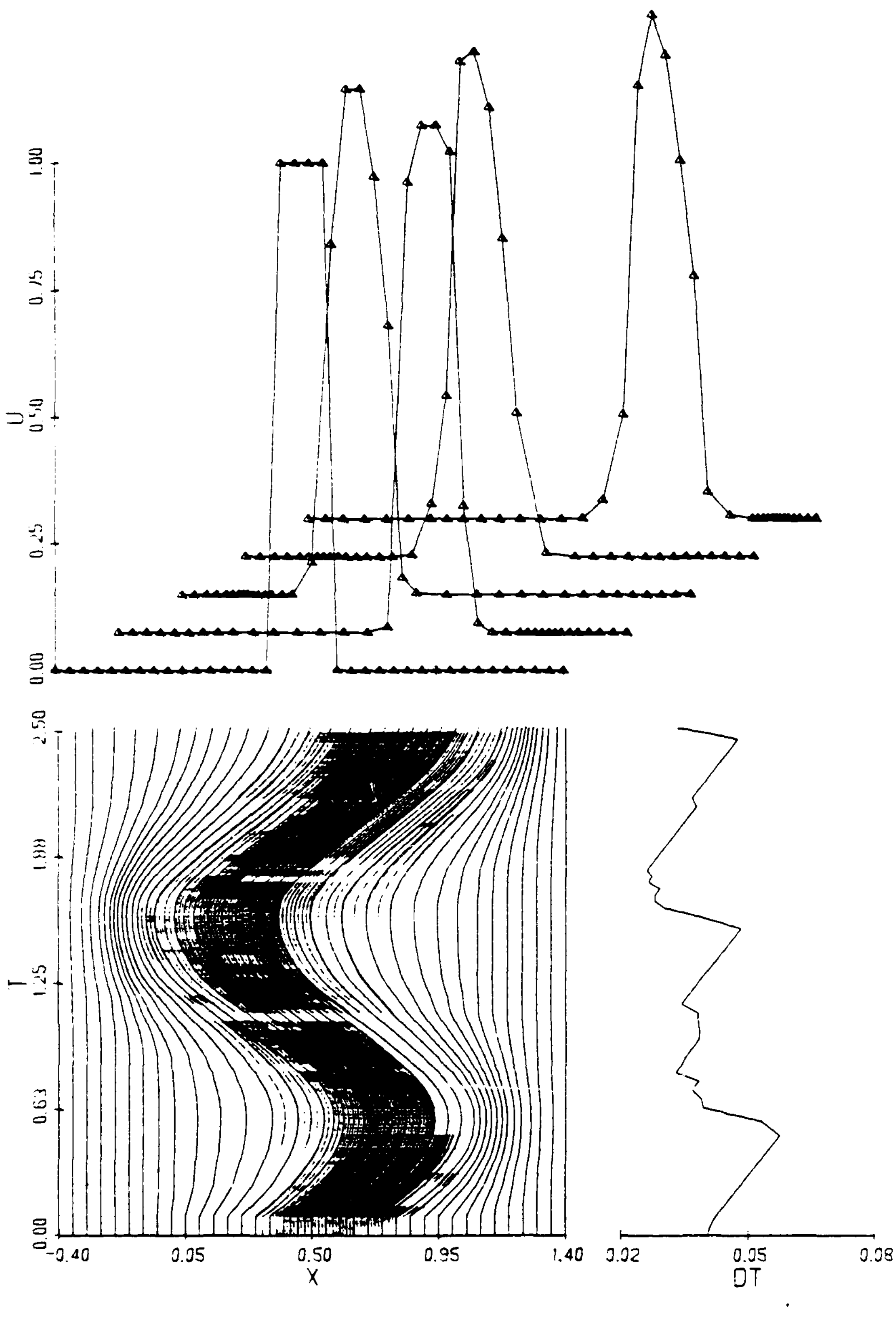

Figure 5. Solutions at t = 0, 0.77, 1.37, 1.91, and 2.50, mesh trajectories, and time step profile for strategy 4 of Example 1.

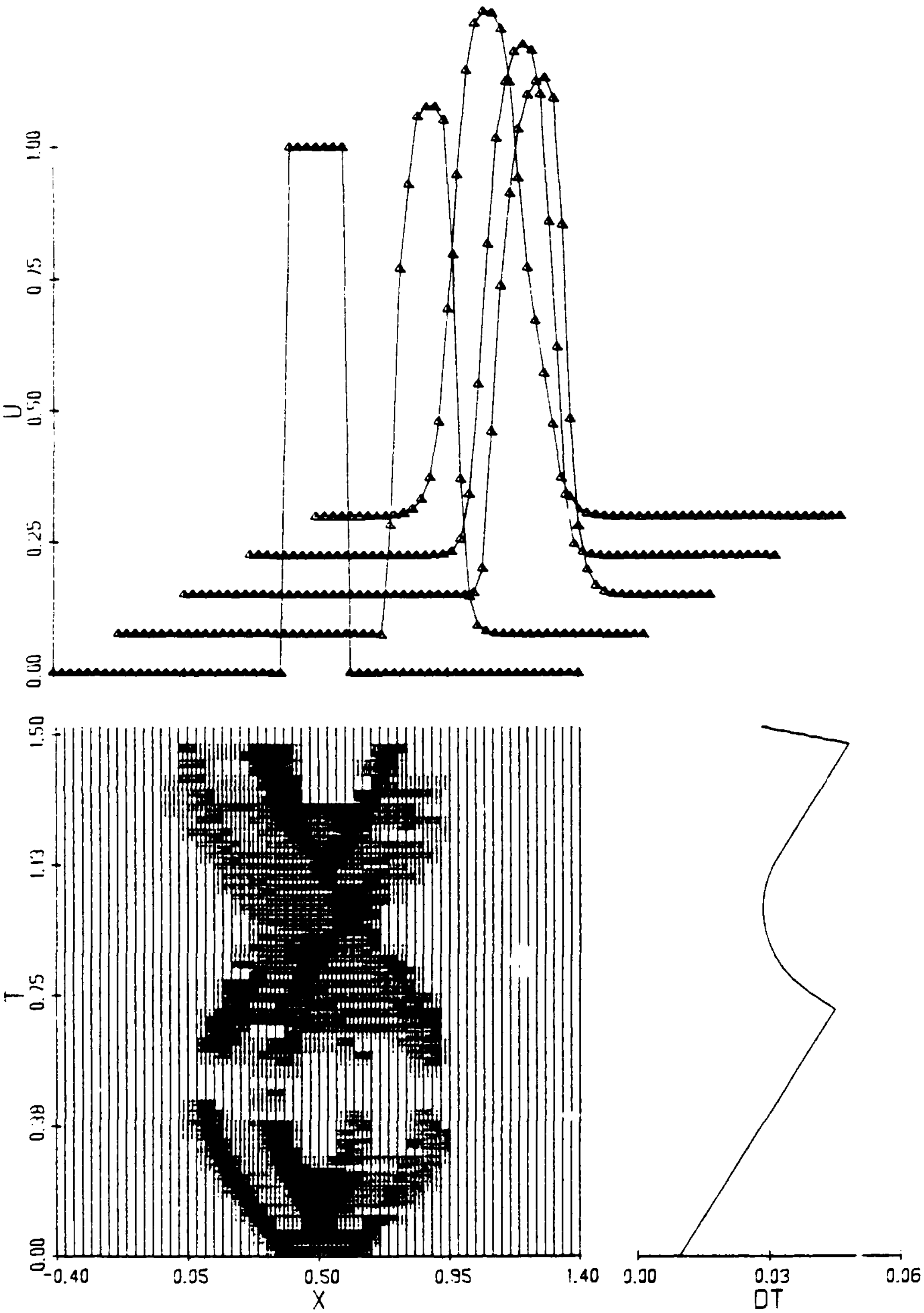

Figure 6. Solutions for u at t = 0, 0.17, 0.48, 0.96, and 1.38, mesh trajectories, and time step profile for strategy 3 of Example 2.

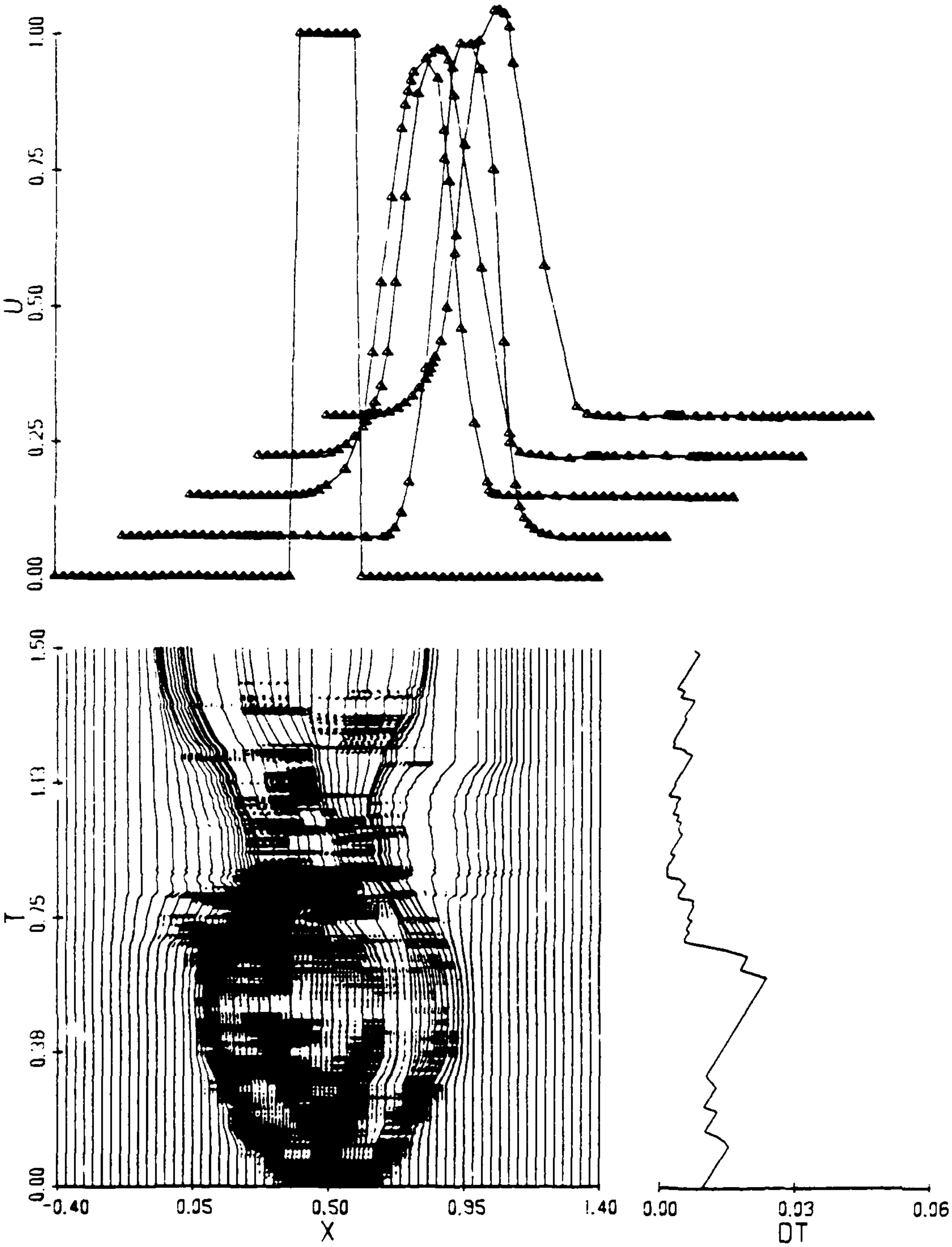

Figure 7. Solutions for u at t = 0, 0.76, 0.95, 1.17, and 1.49, mesh trajectories, and time step profile for strategy 4 of Example 2.

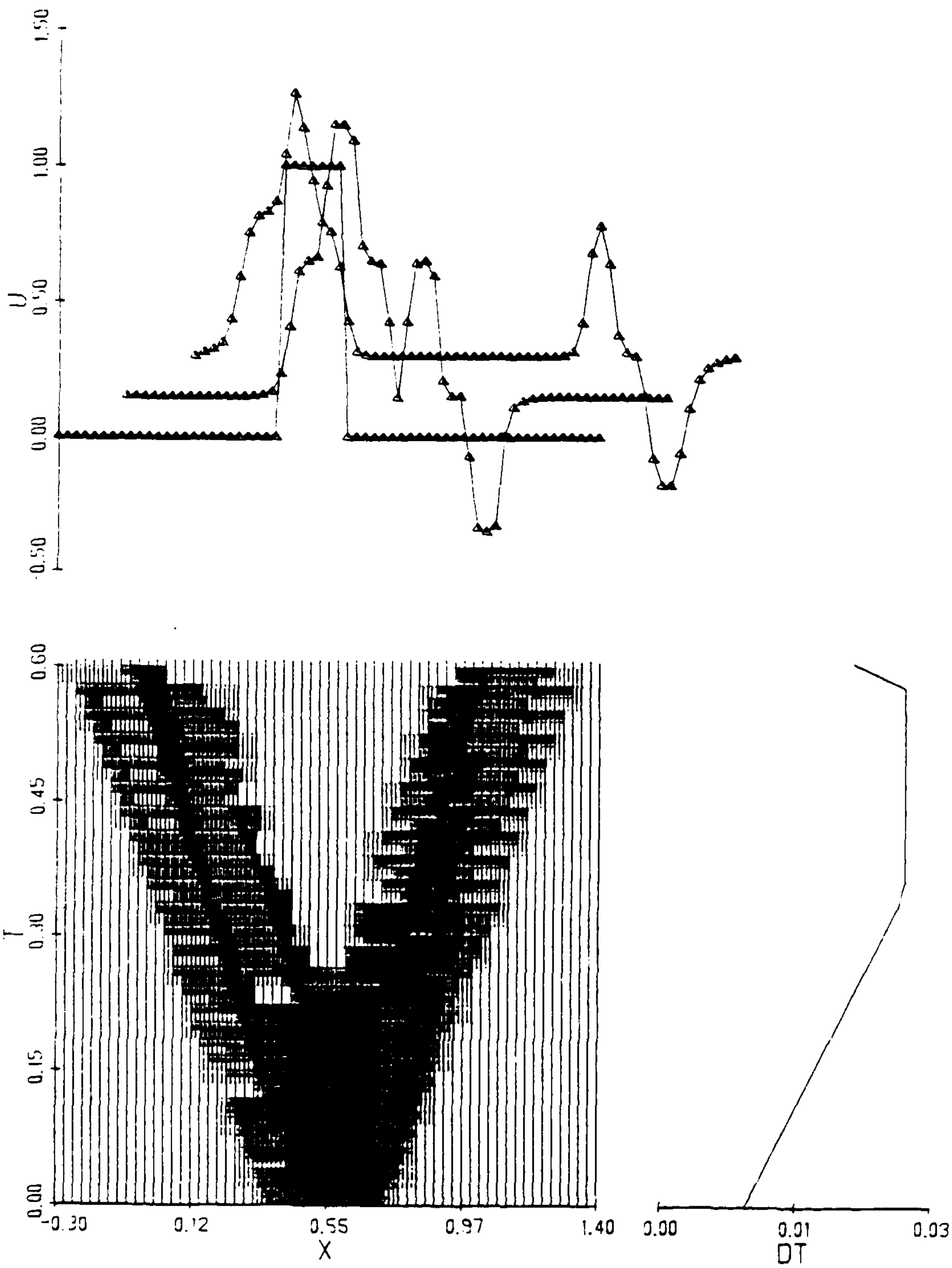

Figure 8. Solutions for u at t = 0, 0.20, and 0.58, mesh trajectories, and time step profile for strategy 3 of Example 3 with error tolerance = 0.005.

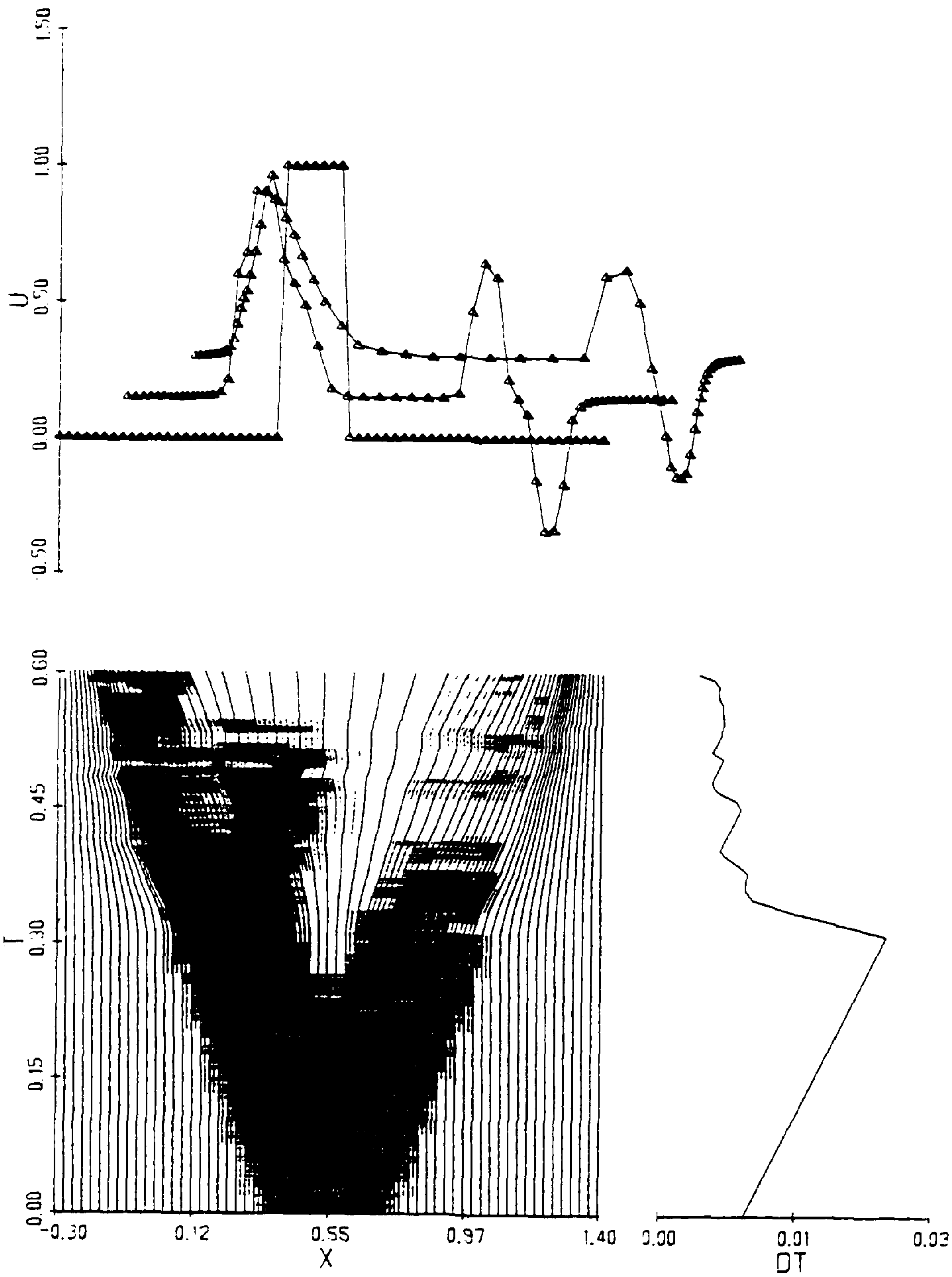

Figure 9. Solutions for u at t = 0, 0.41, and 0.60, mesh trajectories, and time step profile for strategy 4 of Example 3 with error tolerance = 0.005.

TECHNICAL REPORT INTERNAL DISTRIBUTION LIST

	NO. OF COPIES
CHIEF, DEVELOPMENT ENGINEERING DIVISION	
ATTN: SMCAR-CCB-D	1
-DA	1
-DC	1
-DM	1
-DP	1
-DR	1
-DS (SYSTEMS)	1
CHIEF, ENGINEERING SUPPORT DIVISION	
ATTN: SMCAR-CCB-S	1
-SE	1
CHIEF, RESEARCH DIVISION	
ATTN: SMCAR-CCB-R	2
-RA	1
-RM	1
-RP	1
-RT	1
TECHNICAL LIBRARY	5
ATTN: SMCAR-CCB-TL	
TECHNICAL PUBLICATIONS & EDITING SECTION	3
ATTN: SMCAR-CCB-TL	
DIRECTOR, OPERATIONS DIRECTORATE	1
ATTN: SMCWV-OD	
DIRECTOR, PROCUREMENT DIRECTORATE	1
ATTN: SMCWV-PP	
DIRECTOR, PRODUCT ASSURANCE DIRECTORATE	1
ATTN: SMCWV-QA	

NOTE: PLEASE NOTIFY DIRECTOR, BENET LABORATORIES, ATTN: SMCAR-CCB-TL, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST

	NO. OF COPIES
ASST SEC OF THE ARMY RESEARCH AND DEVELOPMENT ATTN: DEPT FOR SCI AND TECH THE PENTAGON WASHINGTON, D.C. 20310-0103	1
ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN: DTIC-FDAC CAMERON STATION ALEXANDRIA, VA 22304-6145	12
COMMANDER US ARMY ARDEC	
ATTN: SMCAR-AEE	1
SMCAR-AES, BLDG. 321	1
SMCAR-AET-O, BLDG. 351N	1
SMCAR-CC	1
SMCAR-CCP-A	1
SMCAR-FSA	1
SMCAR-FSM-E	1
SMCAR-FSS-D, BLDG. 94	1
SMCAR-IMI-I (STINFO) BLDG. 59	2
PICATINNY ARSENAL, NJ 07806-5000	
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-DD-T, BLDG. 305 ABERDEEN PROVING GROUND, MD 21005-5066	1
DIRECTOR US ARMY MATERIEL SYSTEMS ANALYSIS ACTV ATTN: AMXSY-MP ABERDEEN PROVING GROUND, MD 21005-5071	1
COMMANDER HQ, AMCCOM ATTN: AMSMC-IMP-L ROCK ISLAND, IL 61299-6000	1

	NO. OF COPIES
COMMANDER ROCK ISLAND ARSENAL ATTN: SMCRI-ENM ROCK ISLAND, IL 61299-5000	1
DIRECTOR US ARMY INDUSTRIAL BASE ENGR ACTV ATTN: AMXIB-P ROCK ISLAND, IL 61299-7260	1
COMMANDER US ARMY TANK-AUTMV R&D COMMAND ATTN: AMSTA-DDL (TECH LIB) WARREN, MI 48397-5000	1
COMMANDER US MILITARY ACADEMY ATTN: DEPARTMENT OF MECHANICS WEST POINT, NY 10996-1792	1
US ARMY MISSILE COMMAND REDSTONE SCIENTIFIC INFO CTR ATTN: DOCUMENTS SECT, BLDG. 4484 REDSTONE ARSENAL, AL 35898-5241	2
COMMANDER US ARMY FGN SCIENCE AND TECH CTR ATTN: DRXST-SD 220 7TH STREET, N.E. CHARLOTTESVILLE, VA 22901	1
COMMANDER US ARMY LABCOM MATERIALS TECHNOLOGY LAB ATTN: SLCMT-IML (TECH LIB) WATERTOWN, MA 02172-0001	2

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST (CONT'D)

	NO. OF COPIES
COMMANDER US ARMY LABCOM, ISA ATTN: SLCIS-IM-TL 2800 POWDER MILL ROAD ADELPHI, MD 20783-1145	1
COMMANDER US ARMY RESEARCH OFFICE ATTN: CHIEF, IPO P.O. BOX 12211 RESEARCH TRIANGLE PARK, NC 27709-2211	1
DIRECTOR US NAVAL RESEARCH LAB ATTN: MATERIALS SCI & TECH DIVISION	1
CODE 26-27 (DOC LIB) WASHINGTON, D.C. 20375	1
COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MN EGLIN AFB, FL 32542-5434	1
COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MNF EGLIN AFB, FL 32542-5434	1
METALS AND CERAMICS INFO CTR BATTELLE COLUMBUS DIVISION 505 KING AVENUE COLUMBUS, OH 43201-2693	1

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.